

REMARKS

Applicant is in receipt of the Office Action mailed July 14, 2005. Claim 355 has been amended. Claims 285-381 remain pending in the case. Further consideration of the present case is earnestly requested, in light of the following remarks.

Amendments to the claims

Applicant has amended independent claim 355 to include the limitation: *automatically connecting the node to at least one other node in the new graphical program*, to make this claim more consonant with the other independent claims, and thereby simplify issues for Appeal. Note that no new matter has been added, and that no new issues have been raised, as each of the other independent claims already includes the limitation(s) of either automatically creating a node, or automatically connecting a node, in the new graphical program.

Applicant respectfully requests entry of the amendment.

Section 103 Rejections

Claims 370-380 were rejected under 35 U.S.C. 103(a) as being unpatentable over Sojoodi et al (U.S. Patent No. 6,437,805, "Sojoodi") and Yoshida et al (U.S. Patent No. 5,767,853, "Yoshida"). Applicant respectfully disagrees.

Applicant notes that independent claims 370 and 379 each include the limitation that the graphical program is created *automatically*, i.e., *without any user input specifying selection of graphical program nodes and interconnection of graphical program nodes*.

The Examiner asserts that "Sojoodi teaches a system for automatically creating a graphical program..." As argued in the previous two Responses (hereby incorporated by reference), Sojoodi describes *manual* creation of a graphical program which is operable to access capabilities of an object, and specifically does *not* describe automatically creating the graphical program via execution of a program. Applicant notes that the claims as currently written emphasize the fact that the graphical program is *not* generated manually, i.e., is not generated in response to user input specifying selection of graphical program nodes and interconnection of graphical program nodes, but rather is generated *automatically*, i.e., without any user input specifying the plurality of graphical program

nodes or the interconnection of the plurality of graphical program nodes during said creating. Applicant notes that the Background section, page 5, lines 15-22, recites:

As described above, a user typically creates a graphical program within a graphical programming environment by *interactively or manually placing icons or nodes representing the desired blocks of functionality on a diagram, and connecting the icons/nodes together* to represent the data flow of the program. The ability to *programmatically create/edit graphical programs* enables a graphical program to *automatically* be created/edited *without this type of interactive user intervention*. A system and method for programmatically creating/editing graphical programs is described herein.

It is this type of graphical program creation that is disclosed by Sojoodi. In direct contrast, the Summary, page 6, lines 4-9 recites:

The ability to programmatically create/edit graphical programs enables applications and tools to *automatically* create or modify a graphical program or a portion of a graphical program. In the preferred embodiment, any operation which a user may perform while interactively assembling or editing a graphical program (e.g., connect two objects, change the position of an object, change the color of an object, etc.) may be performed programmatically.

Similarly, regarding client/server embodiments, page 8, lines 1-4 recites:

As noted above, the program which calls the API in order to *automatically* create or edit a graphical program is referred to herein as the client program. The program which performs the create/edit operations on the graphical program is referred to herein as the server program.

The Examiner has repeatedly asserted that Sojoodi discloses a client program (executing in the computer system) performing API calls to programmatically create a graphical program cites col. 5, lines 28-30, in. However, as noted in the previous Response, the cited portion actually reads:

The present invention comprises a system and method for creating a graphical program, wherein the graphical program is operable to access capabilities of an object.

Nowhere does the cited portion (or any other portion of Sojoodi) refer to “automatically” creating a graphical program. Rather, Sojoodi discloses (Abstract):

During creation of the graphical program, *the user* operates to place an object node in the graphical program, wherein the object node is operable to access capabilities of the object. This preferably includes *the user*

arranging on the screen the graphical program, including the object node and various other nodes, and connecting the various nodes to create the graphical program. (*emphasis added*)

Clearly, in Sojoodi's system, the *user* selects, arranges, and connects the nodes. Similarly, the Examiner cites col. 4, lines 43-53, in asserting that Sojoodi discloses a server program operable to receive the client program calls to programmatically create a graphical program and operable to perform the respective operations. The cited portion actually reads:

The notion of object technology relates to an application program or object using capabilities or services of an object. For example, object technology includes the notion of "servers" "exporting" their "objects" for use by "clients." A server is a software application which makes available its classes to be accessed from another application, referred to as a client. A client instantiates objects from the classes in the type libraries. In addition, the term "object" also includes various other types of components or objects, including applications, which offer services or functionality that can be accessed by a client.

Nowhere does the cited portion (or any other portion) refer to a server program operable to receive client program calls to *automatically* create a graphical program based on received information and operable to perform the respective operations, i.e., to automatically create the graphical program.

Further arguments and citations regarding Sojoodi's teaching of *manual* creation of a graphical program were provided in the previous Responses, which were incorporated by reference above.

Thus, Applicant respectfully submits that Sojoodi does not disclose automatically creating the graphical program without user input specifying or guiding the inclusion and interconnection of graphical program nodes, and more specifically fails to teach a client program performing API calls to automatically create a graphical program, and in response, a server program automatically creating a plurality of graphical program nodes in the new graphical program, and automatically interconnecting the plurality of graphical program nodes in the new graphical program.

The Examiner states that "Sojoodi does not teach creating the new program from a program without any user input specifying the nodes and interconnection of the plurality of node" [sic]. Applicant submits that this statement is itself misleading, in that

the particular wording “creating the new program from a program” implies that a new program is necessarily created *based on* another program, i.e., that the new program is necessarily created by converting or translating another program, which is *not* an accurate description or characterization of Applicant’s invention as claimed.

The Office Action asserts that Yoshida corrects the deficiencies of Sojoodi. Applicant respectfully disagrees. Yoshida is directed to invocation of an application in response to user-designation of a data file based on past user operations, e.g., based on the frequency of past associations between the application and the data file, based on the last application that used the data file, etc. Nowhere does Yoshida teach or suggest automatically creating a graphical program. More specifically, Yoshida fails to teach automatic creation of a graphical program without any user input specifying the nodes and interconnection of the plurality of node, nor a server program executable to perform such automatic creation the graphical program.

Applicant notes that Yoshida never mentions automatic creation of a graphical program in response to received information, nor does Yoshida even mention a graphical program at all. In other words, as argued above, neither Sojoodi nor Yoshida teaches or suggests automatic creation of a graphical program by *any* means. Thus, for at least the reasons provide above, Applicant submits that neither Sojoodi nor Yoshida, taken singly or in combination, teaches or suggests all the features and limitations of claims 370 and 379, and so claims 370 and 379, and those claims respectively dependent therefrom, are patentably distinct and non-obvious over Sojoodi and Yoshida, and are thus allowable.

Claims 285-369 and 381 were rejected under 35 U.S.C. 103(a) as being unpatentable over McDonald et al (USP 5,6966,532, “McDonald”), Sojoodi, and Yoshida. Applicant respectfully disagrees.

McDonald is directed to generating a graphical program (or portion) by programmatically selecting *pre-existing code templates* based on user-selected graphical user interface controls and inserting the code templates into a graphical program, which is not at all the same as the present invention as represented in claim 285. The Office Action asserts that McDonald teaches “the creating of a first program, when executing, programmatically creating a new graphical program” [sic], citing col. 3, lines 61-63.

However, Applicant notes that the cited portion of McDonald actually reads: “The present invention comprises a computer-implemented system and method for automatically generating graphical code in a graphical programming system.” Applicant submits that there are numerous possible ways to automatically generate graphical code in a graphical programming system, and that the method disclosed by McDonald is not at all the same as that claimed in the present application. Nowhere does McDonald teach or suggest “creating a first program, wherein the first program is executable to automatically create a new graphical program”, nor executing the first program to automatically create the new graphical program, *including automatically creating a plurality of graphical program nodes in the new graphical program, and automatically interconnecting the plurality of graphical program nodes in the new graphical program...* wherein said automatically creating the new graphical program creates the new graphical program without any user input specifying the plurality of graphical program nodes or the interconnection of the plurality of graphical program nodes program during said creating. The graphical code templates selected in McDonald have preconfigured nodes and interconnections, and hence McDonald does not perform, and has no reason to perform, any automatic creation or interconnection of nodes.

Applicant notes that each of independent claims 285, 333, 353, 355, 362, and 381 includes automatically creating a node, or automatically creating a connection between nodes, or both, neither of which is taught or suggested by McDonald, Sojoodi, or Yoshida.

Additionally, Applicant respectfully submits that neither McDonald, Sojoodi, nor Yoshida provides a motivation to combine, and so the attempted combination of Sojoodi and Yoshida, and of McDonald, Sojoodi, and Yoshida is improper. For example, nowhere does Sojoodi or Yoshida suggest the desirability or benefit of automatically creating a graphical program, i.e., without user input selecting the nodes and connections between the nodes. Applicant notes that the only motivation to combine indicated by the Examiner is “to automatically translate graphical program from one programming language to another programming language [sic]”, which Applicant submits is not a feature of Applicant’s invention as claimed, and so is an improper motivation to combine.

Similarly, nowhere does McDonald indicate the desirability or benefit of automatically creating a graphical program in the manner claimed, e.g., by automatically creating and connecting one or more nodes without user input specifying the nodes or connections.

Thus, for at least the reasons provide above, Applicant submits that neither McDonald, Sojoodi, nor Yoshida, taken singly or in combination, teaches or suggests all the features and limitations of claims 285, 333, 353, 355, 362, and 381, and so claims 285, 333, 353, 355, 362, and 381, and those claims respectively dependent therefrom, are patentably distinct and non-obvious over McDonald, Sojoodi and Yoshida, and are thus allowable.

Removal of the section 103 rejection of claims 285-381 is respectfully requested.

Applicant also asserts that numerous ones of the dependent claims recite further distinctions over the cited art. However, since the independent claims have been shown to be patentably distinct and non-obvious, a further discussion of the dependent claims is not necessary at this time.

CONCLUSION

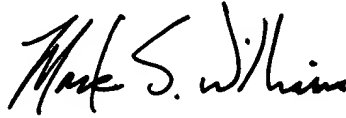
Applicant submits the application is in condition for allowance, and an early notice to that effect is requested.

If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert & Goetzel PC Deposit Account No. 50-1505/5150-37301/JCH.

Also enclosed herewith are the following items:

☒ Return Receipt Postcard

Respectfully submitted,



Mark S. Williams
Reg. No. 50,658
AGENT FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800
Date: 8-29-05 JCH/MSW